



⑯ BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

⑫ Offenlegungsschrift  
⑯ DE 44 06 094 A 1

⑯ Int. Cl. 6:  
G 06 F 9/46

DE 44 06 094 A 1

⑯ Aktenzeichen: P 44 06 094.7  
⑯ Anmeldetag: 25. 2. 94  
⑯ Offenlegungstag: 31. 8. 95

⑯ Anmelder:  
Munz, Heinrich, 88368 Bergatreute, DE; Leibinger, Josef, 88250 Weingarten, DE

⑯ Vertreter:  
Riebling, P., Dipl.-Ing. Dr.-Ing., Pat.-Anw., 88131 Lindau

⑯ Erfinder:  
gleich Anmelder

⑯ Für die Beurteilung der Patentfähigkeit in Betracht zu ziehende Druckschriften:  
DE 37 26 659 A1  
DE 33 35 549 A1  
GARROW, Robert;  
et.al.: 16-bit single-board computermaintains 8-bit

family ties. In: Electronics, Oct. 12, 1978, H.21, S.105-110;  
N.N.: Interrupt Controller Diagnostic Wrap Arrangement. In: IBM Technical Disclosure Bulletin, Vol. 29, No.9, Febr. 1987, S.3871, 3872;  
HALL, J.D.;  
et.al.: Multiple Interrupt Registration. In: IBM Technical Disclosure Bulletin, Vol.26, No.1, June 1983, S.226,227;  
HIGDON, J.M.;  
WRIGHT, F.D.:Programmable, Expandable Interrupt Controller. In: IBM Technical Disclosure Bulletin, Vol.24, No.5,Okt.1981,S.2303,2306;

Prüfungsantrag gem. § 44 PatG ist gestellt

⑯ Verfahren und Vorrichtung zum Echtzeitbetrieb eines Prozessors

⑯ Die Erfindung schlägt ein Verfahren und eine Vorrichtung zum Echtzeitbetrieb eines Prozessors unter einem nichtechtzeitfähigen Betriebssystem vor, wobei die bestehenden externen Hardware-Interrupt-Quellen unter Umgehung der bestehenden Interrupt-Hard- und Software direkten Zugriff auf den nicht-maskierbaren Interrupt (NMI) des Prozessors erhalten.

DE 44 06 094 A 1

Die Erfindung betrifft ein Verfahren und eine Vorrichtung zum Echtzeit-Betrieb eines Prozessors nach dem Oberbegriff des Patentanspruchs 1.

In dem Zeitschriftenartikel Schramm, Dr. Mathias; Siering, Peter: Blitz ohne Donner, Hardware-Interrupts unter Windows, c't 1994, Heft 2, Seite 188–191, ist angegeben, daß es unter dem Betriebssystem Windows praktisch keinen Echtzeit-Betrieb im Sinne der DIN-Definition gibt.

Im Sinne dieser DIN-Definition bezeichnet man damit die Reaktionen eines Computer-Systems auf externe Unterbrechungen (Interrupts) innerhalb einer vorgegebenen (kurzen) Zeit. Ein wichtiger Begriff, der die Leistungsfähigkeit eines Systems charakterisiert, ist die Interrupt-Latenzzeit. Diese typische Systemkonstante beschreibt die maximale Zeit, die zwischen externen Anforderung (Interrupt-Request) und Reaktion des Steuerungsprogrammes vergehen kann. Sie liegt in der Meßtechnik je nach spezieller Anforderung im Bereich von einigen Mikro- bis zu einigen Millisekunden.

Setzt eine externe Hardware eine Interrupt-Leitung des 8259 A-Programmable-Interrupt-Controllers auf logisch "high", so sendet dieser die zugeordnete Vektornummer (0 bis 255) an den Prozessor, der seinerseits den Aufruf der jeweiligen Behandlungsroutine initiiert. Unter MSDOS dauert dieser Vorgang 12 bis 15 Mikrosekunden; unter Windows jedoch gelangt die Interrupt-Routine nicht direkt zum Ablauf. Teile des virtuellen Interrupt-Controller-Treibers (VPICD) werden zur Initialisierung aufgerufen, was die Interrupt-Latenzzeit typischerweise auf circa 50 Mikrosekunden erhöht. Das Eintreffen höher priorisierter Interrupts zum Beispiel von einem Timer oder Netzwerk kann den Ablauf der Initialisierung unterbrechen, so daß bei Systemen mit hoher Belastung diese Zeiten massiv überschritten werden können.

Es war bisher also nicht möglich, unter dem Betriebssystem Windows eine konstante Interrupt-Latenzzeit zu definieren, weil das System unterschiedlich schnell auf unterschiedliche Interrupt-Anforderungen reagierte.

Es ist bisher so, daß die externen Hardware-Interrupt-Quellen durch einen Software-Befehl maskiert werden können, was bedeutet, daß diese Interrupt-Quellen nicht bearbeitet werden, wenn dies von der Software verlangt wird. Dies führt software-bedingt zu einer nicht vorhersagbaren Interrupt-Latenzzeit, was bisher einen Echtzeit-Betrieb unter dem Betriebssystem Windows verhinderte.

Dieser Nachteil gilt im übrigen auch für alle anderen nicht-echtzeitfähigen Betriebssysteme, wie z. B. DOS, UNIX, Windows-NT, und andere, wie z. B. MacIntosh.

Aufgabe der vorliegenden Erfindung ist es deshalb, ein Verfahren und eine zur Ausführung des Verfahrens geeignete Vorrichtung vorzuschlagen, mit dem es gelingt, eine maximale Interrupt-Latenzzeit zu garantieren.

Die Lösung der gestellten Aufgabe erfolgt durch die technische Lehre nach dem Gegenstand des Anspruchs 1 und die zur Ausführung des Verfahrens notwendige Vorrichtung.

Die Grundidee besteht darin, einem PC, welcher unter einem "normalen", nicht echtzeitfähigen, Betriebssystem (BS) läuft, (in diesem Falle MS-Windows 3.1), garantierte Echtzeitfähigkeit (im Mikro-Sekunden-Bereich) zu verleihen, um damit Maschinensteuerungen sozusagen im Hintergrund von Windows ablaufen zu

lassen.

Damit ist nun erstmals sichergestellt, daß man die Reaktion eines Systems auf einen Hardware-Interrupt genau vorhersagen kann, so daß ein derartiges System nun erfindungsgemäß im Sinne DIN-Definition echtzeitfähig ist.

Die Interrupt-Latenzzeit kann sich zwar je nach Prozessor-System und dem verwendeten Betriebssystem ändern, sie wird aber für einen bestimmten Prozessor, der unter einem bestimmten Betriebssystem läuft, stets vorhersagbar sein.

Erfindungsgemäß wird dabei der sogenannte NMI (nonmaskable interrupt) verwendet, um zur Lösung der gestellten Aufgabe zu kommen.

Der NMI wird also verwendet, um unmittelbar und vorhersehbar auf den Prozessor zugreifen zu können.

Das Betriebssystem kann nämlich die normalen Hardware-Interrupt-Quellen einer 80 × 86 CPU unbekannt lange sperren, insbesondere beim Durchlaufen des BIOS. Erfindungsgemäß ist aber der NMI einer 80 × 86 CPU auf dem PC-Bus über das Signal IOCHCK (Input-Output-Channel-Check) zugänglich.

Das heißt, erfindungsgemäß wird als Interrupt-Quelle für Echtzeitanforderungen der sogenannte NMI verwendet, der bei einer AT-Busbelegung auch mit dem Input-Output-Channel-Check (IOCHCK) bezeichnet wird.

Das Signal ist jedoch nicht nur auf dem AT-Bus oder ISA-Bus vorhanden, sondern auch auf den älteren PC-Bus-Systemen.

Erfindungsgemäß wird also dieses Signal verwendet, um einen Echtzeitbetrieb der CPU durchführen zu können.

Die Erfindung ist jedoch nicht auf die beschriebenen Bus-Systeme beschränkt, sondern kann für beliebige Bus-Systeme verwendet werden, wie z. B. auch den VESA-Bus oder den Local-Bus.

Selbst wenn man einen geschlossenen Rechner ohne externes Bus-System hat, kann man dieses Signal verwenden, um eine Echtzeitfähigkeit des Prozessors herzuführen.

Mit der vorliegenden Erfindung ist es nun also zum ersten Mal möglich, die sogenannten PC-Prozessoren für Steuerungsanwendungen zu verwenden, was bisher einer eigenen Rechnerfamilie, z. B. des Typs 68000, vorbehalten war.

Man hat bisher nicht die Möglichkeit in Betracht gezogen, andere Hardware-Interrupt-Quellen zur Unterbrechung des Prozessors zu verwenden, als die bisher bekannten Quellen, und erst die Erfindung sieht vor, daß man eigentlich eine nicht als Hardware-Interrupt-Quelle verwendbare Signalleitung, die eigentlich nur für interne Zwecke verwendet wird, als Hardware-Interrupt-Quelle verwendet.

Eine Vorrichtung zur Ausführung des erfindungsgemäßen Verfahrens sieht vor, daß man an die Signalleitung des NMI (IOCHCK) direkt eine Hardware-Unterbrechungsquelle anschließt. Eine derartige Hardware-Unterbrechungsquelle kann z. B. eine beliebige Steckkarte sein, über welche eine Echtzeit-Datenverarbeitung gesteuert wird.

In einer Weiterbildung der erfindungsgemäßen Vorrichtung ist es vorgesehen, daß die Hardware-Interrupt-Quellen über einen sogenannten programmierbaren Verteiler so gesteuert werden, daß in Abhängigkeit von der Programmierung des Verteilers entweder die normalen, externen Hardware-Interrupt-Quellen Zugriff auf den Prozessor erhalten oder wahlweise — in Abhän-

gigkeit vom Programm — das neuartige NMI-Signal aktiviert wird und unmittelbar auf den Prozessor zugreift.

Ein derartiger programmierbarer Verteiler wählt also aus einer Anzahl von externen Hardware-Interrupt-Quellen diejenigen heraus, die Zugriff auf den Prozessor erhalten sollen.

Die erfindungsgemäße NMI wird dann aus einem oder mehreren dieser Hardware-Interrupt-Quellen gebildet, indem er quasi direkt galvanisch mit einer oder mehreren dieser Hardware-Interrupt-Quellen verbunden wird.

Statt der galvanischen Verbindung ist bevorzugt eine softwaremässige Umschaltung von Hardware-Interrupt-Quellen über ein logisches Netzwerk auf diesen NMI vorgesehen.

Es kann hierbei vorgesehen sein, den programmierbaren Verteiler auch mit einer bestimmten Speicherfähigkeit zu versehen, um die eingehenden Hardware-Interrupt-Signale auf einen bestimmten logischen Pegel für eine gewisse Zeit zu halten und abzuspeichern.

Demgemäß liegen die auf dem Bus zur Verfügung stehenden Interrupt-Leitungen sowohl am Interrupt-Controller des Motherboards, als auch an einem IRQ-Router (Interrupt-Quellen-Router) an. Dieser ist in Form einer Zusatzkarte realisiert. Per Software kann bestimmt werden, ob und welche der auf dem Bus anstehenden Interrupts in einem NMI (Non-Maskable-Interrupt) "umgeroutet" werden sollen. Der vom IRQ-Router generierte NMI gelangt über das ebenfalls auf dem Bus vorhandene Signal IOCHCK zum 80 x 86-Prozessor.

Der Erfindungsgegenstand der vorliegenden Erfindung ergibt sich nicht nur aus dem Gegenstand der einzelnen Patentansprüche, sondern auch aus der Kombination der einzelnen Patentansprüche untereinander. Alle in den Unterlagen, einschließlich der Zusammenfassung, offenbarten Angaben und Merkmale, insbesondere die in den Zeichnungen dargestellte räumliche Ausbildung werden als erfindungswesentlich beansprucht, soweit sie einzeln oder in Kombination gegenüber dem Stand der Technik neu sind.

Im folgenden wird die Erfindung anhand von lediglich einem Ausführungsweg darstellenden Zeichnungen näher erläutert. Hierbei gehen aus den Zeichnungen und ihrer Beschreibung weitere erfindungswesentliche Merkmale und Vorteile der Erfindung hervor.

In der Abbildung ist schematisiert ein Blockschaltbild eines sogenannten IRQ-Routers (Interrupt-Quellen-Verteilers) gezeigt.

Dabei ist in der unteren Hälfte der Zeichnung (unterhalb der gestrichelten Linie) die normalerweise in einem Prozessorsystem schon vorhandene Interrupt-Hardware gezeigt, und in der oberen Hälfte die erfindungsgemäße zusätzliche Hardware in Form des IRQ-Routers.

In einem PC-Bus 2 werden eine Anzahl von externen Hardware-Interrupt-Quellen IRQ ... (vergleiche in der Abbildung links) geführt.

Diese Hardware-Interrupt-Quellen 1 werden zu einem Teil über einen ersten Unterbrecher-Controller 3 einem zweiten Unterbrecher-Controller 4 zugeführt oder sie werden wahlweise dem zweiten Unterbrechungs-Controller zugeführt.

Der zweite Unterbrechungs-Controller 4 erzeugt ein Signal, welches über das sogenannte maskierbare Interrupt-Pin 6 dem Prozessor 7 zugeführt wird. Auf diese Weise kann ein Interrupt am Prozessor 7 erzeugt werden, welcher allerdings durch einen Software-Befehl abgeschaltet werden kann.

Erfindungsgemäß ist nun in einer ersten nicht näher dargestellten Schaltung vorgesehen, daß eine externe Hardware-Interrupt-Quelle unmittelbar mit der Leitung 10 verbunden ist, welche Leitung mit dem sogenannten NMI gekoppelt ist. Diese Signalleitung führt eine unmittelbare Unterbrechung am Prozessor herbei, unabhängig von einem anliegenden Software-Befehl.

In einer Weiterbildung einer Vorrichtung nach der Erfindung ist gemäß der Abbildung vorgesehen, daß die Hardware-Interrupt-Quellen aus dem PC-Bus 2 herausgeführt werden und einem sogenannten IRQ-Router (Verteiler 8) zugeführt werden.

Dieser Verteiler 8 ist programmierbar und entscheidet, welche Hardware-Interrupt-Quelle sich auf die Leitung 10 mit dem nicht-maskierbaren Interrupt 9 aufschalten darf.

Sobald auf dieser Leitung 10 ein Signal erscheint, wird der Prozessor unbedingt unterbrochen, unabhängig davon, ob per Software ein Interrupt gesperrt ist oder nicht.

Das bedeutet, entsprechend einem Signal auf der Leitung 10 ist also der Prozessor 7 echtzeitfähig, weil er zu beliebigen Zeitpunkten ein Interrupt durchführen kann.

## 25 Bezugszeichenliste

- 1 Hardware-Interrupt-Quellen (extern)
- 2 PC-Bus
- 3 Unterbrechungskontroller
- 4 Unterbrechungskontroller
- 5 Hardware-Interrupt-Quelle (intern)
- 6 maskierbarer Interrupt-Pin
- 7 Prozessor
- 8 Verteiler
- 9 nicht-maskierbarer Interrupt
- 10 Leitung

## Patentansprüche

1. Verfahren zum Echtzeitbetrieb eines Prozessors unter einem nicht-echtzeitfähigen Betriebssystem, dadurch gekennzeichnet, daß die bestehenden, externen Hardware-Interrupt-Quellen (1) unter Umgehung der bestehenden Interrupt-Hard- und Software (3, 4, 5) direkten Zugriff auf den nicht-maskierbaren Interrupt (NMI) (9) des Prozessors (7) erhalten.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß der Zugriff auf den NMI (9) des Prozessors (7) über die IOCHCK-Signalleitung (10) des PC-Bus erfolgt.

3. Vorrichtung zur Ausübung des Verfahrens nach den Ansprüchen 1 und 2, dadurch gekennzeichnet, daß ein Verteiler (8) vorgesehen ist, der aus den externen Hardware-Interrupt-Quellen (1) eine oder mehrere auswählt, welche Zugriff auf den NMI (9) erhalten sollen.

4. Vorrichtung nach Anspruch 3, dadurch gekennzeichnet, daß der Verteiler (8) programmierbar ist und daß per Software und/oder Hardware bestimmt wird, ob und welche der auf den Bus anstehenden Hardware-Interrupts (1) den NMI (9) bilden.

5. Vorrichtung nach einem der Ansprüche 3 und 4, dadurch gekennzeichnet, daß der Verteiler eine Speicherwirkung aufweist, um wahlweise die Eingangs- und/oder Ausgangssignale für eine gewisse Zeit zwischenspeichern.

6. Vorrichtung nach einem der Ansprüche 3—5, durch gekennzeichnet, daß unter Umgehung des Verteilers (8) direkt ein oder mehrere Hardware-Interrupt-Quellen mit dem NMI (9) verbunden werden können.

5

7. Vorrichtung nach einem der Ansprüche 3—6, durch gekennzeichnet, daß der Verteiler (8) direkt auf dem Motherboard integriert ist.

8. Vorrichtung nach einem der Ansprüche 3—6, durch gekennzeichnet, daß sich der Verteiler (8) auf einer Steckkarte, einem Aufsteckmodul oder einem Ansteckmodul zum Anstecken an Schnittstellen befindet.

10

---

Hierzu 1 Seite(n) Zeichnungen

15

20

25

30

35

40

45

50

55

60

65

**- Leerseite -**

